



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원 번호 : 10-2002-0074641
Application Number

출원 년 월 일 : 2002년 11월 28일
Date of Application NOV 28, 2002

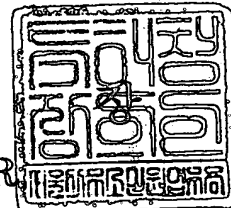
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 07 29 일
 년 월

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0007
【제출일자】	2002.11.28
【국제특허분류】	G06F
【발명의 명칭】	S M I L 을 사용하여 제작된 멀티미디어 콘텐츠, 그 제작 방법 및 그 재생 방법
【발명의 영문명칭】	Multimedia contents using Synchronized Multimedia Integration Language, method for making and reproducing the same
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	1999-009556-9
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2000-002816-9
【발명자】	
【성명의 국문표기】	홍진우
【성명의 영문표기】	HONG, Jin Woo
【주민등록번호】	730215-1109318
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 1042-11 B04호
【국적】	KR
【발명자】	
【성명의 국문표기】	배대규
【성명의 영문표기】	BAE, Dae Gyu
【주민등록번호】	700820-1772611

【우편번호】	442-756
【주소】	경기도 수원시 팔달구 원천동 548 원천주공2단지 202동 201호
【국적】	KR
【발명자】	
【성명의 국문표기】	성현아
【성명의 영문표기】	SUNG, Hyun Ah
【주민등록번호】	770205-2011111
【우편번호】	151-895
【주소】	서울특별시 관악구 신림9동 1518-17호 4층 5/5
【국적】	KR
【발명자】	
【성명의 국문표기】	이주한
【성명의 영문표기】	LEE, Ju Han
【주민등록번호】	731223-1053114
【우편번호】	442-192
【주소】	경기도 수원시 팔달구 우만2동 67-8번지 303호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이영필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	13 면 13,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	28 항 1,005,000 원
【합계】	1,047,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

SMIL을 사용하여 제작된 멀티미디어 콘텐츠, 그 제작 방법 및 재생 방법이 개시된다.

본 발명에 따라 SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠는 변수를 초기화하는 속성을 가진 제1 엘리먼트; 및 상기 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성을 가진 제2 엘리먼트를 포함한다. 이에 의해, 변수의 연산에 따른 메모리 기능을 구비함으로써 서버와의 접속없이도 콘텐츠만으로 사용자 입력에 대한 결과를 누적하거나 저장하여 보여줄 수 있다. 다시 말해, 프로그래밍 작업을 하지 않고 SMIL에 새롭게 정의된 속성을 사용하여 간단하게 메모리 기능을 구현할 수 있다.

【대표도】

도 4

【명세서】

【발명의 명칭】

SMIL을 사용하여 제작된 멀티미디어 콘텐츠, 그 제작 방법 및 그 재생 방법 {Multimedia contents using Synchronized Multimedia Integration Language, method for making and reproducing the same}

【도면의 간단한 설명】

도 1은 본 발명의 바람직한 실시예에 따른 멀티미디어 콘텐츠를 재생하는 장치의 블록도,

도 2는 본 발명의 바람직한 실시예에 따른 SMIL 콘텐츠의 개략도,

도 3은 도 2의 프레임 구성을 위한 SMIL 콘텐츠의 개략도,

도 4는 도 2의 SMIL 콘텐츠의 일 예,

도 5는 도 2의 SMIL 콘텐츠의 다른 예,

도 6은 도 2의 SMIL 콘텐츠의 또 다른 예,

도 7은 도 2의 SMIL 콘텐츠의 또 다른 예,

도 8은 도 4 내지 도 7에 구현된 변수의 메모리 기능에 의해 저장된 변수를 활용하기 위한 예,

도 9는 변수의 연산에 대한 처리 조건이 추가된 예,

도 10은 본 발명에 따라 함수 기능이 추가된 일 예,

도 11은 본 발명에 따라 함수 기능이 추가된 다른 예,

도 12는 본 발명의 일 실시예에 따라 속성이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트,

도 13은 본 발명의 다른 실시예에 따라 속성이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트,

도 14는 본 발명의 일 실시예에 따라 함수 기능이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<15> 본 발명은 SMIL(Synchronized Multimedia Integration Language)에 관한 것으로, 보다 상세하게는 SMIL을 사용하여 제작된 멀티미디어 콘텐츠, 그 제작 방법 및 재생 방법에 관한 것이다.

<16> SMIL은 멀티미디어의 레이아웃이나 프리젠테이션을 위해 W3C(World Wide Web Consortium)이 제창하고 있는 XML 기반의 마크업 언어로서, 오디오, 비디오, MIDI(Musical Instrument Digital Interface), 텍스트, 이미지 등 다양한 타입의 독립적인 미디어 데이터들을 하나의 동기화된 멀티미디어 콘텐츠로 통합할 수 있으며, 각 미디어 데이터들의 재생 시간의 조절, 화면 상의 레이아웃 조정 및 분할을 할 수 있다.

<17> SMIL은 HTML과 같이 텍스트 에디터로 작성이 가능하고 엘리먼트의 갯수도 많지 않으며 그 문법이 XML 형식에 따르므로 쉽게 배울 수 있어, SMIL을 사용하면 보다 쉽게 멀티미디어 콘텐츠를 제작할 수 있다.

<18> 멀티미디어 콘텐츠는 교육용 콘텐츠로 많이 사용된다. 가령, 선택형 문제풀이를 위한 교육용 콘텐츠를 SMIL로 제작한다면 사용자 입력을 받아서 정답인지 여부를 확인하여 그 결과를 EventBase나 SyncBase를 이용하여 표시할 수 있다. 그 예는 다음과 같다.

<19>

```
<par>
  <image id='img1'.../>
  <image id='img2'.../>
  <image id='img3'.../>
  <image id='img4'.../>
  <text id='rightAnswer'...
    begin='img1.activateEvent'
    .../>
  <text id='wrong'...
    begin='img2.activateEvent;
    img3.activateEvent;
    img4.activateEvent'.../>
```

<20> 위 예제에 따르면, 식별자 img1을 갖는 이미지가 정답인 경우를 나타내고 있다. 사용자가 그 이미지를 클릭하면 대응하는 이벤트가 발생되어 정답, 즉 식별자 rightAnswer를 가진 텍스트가 표시되고 나머지 정답이 아닌 이미지, 즉 식별자 img2, img3, img4를 갖는 이미지를 클릭하면 대응하는 이벤트가 발생되어 오답, 즉 식별자 wrong을 가진 텍스트가 표시된다.

<21> 그러나, SMIL에 따르면 학습 및 평가 결과가 어느 정도인지를 알아보기 위해 지금까지의 점수를 계산한다거나, 각 평가 항목마다 중요도에 따라 배점을 달리한다거나 하는 기능을 구현할 수 없다. 다시 말해, SMIL은 변수의 값이 저장되는 메모리 기능을 지원하지 않기 때문이다. 변수의 값을 저장할 수 있도록 하기 위해 프로그래밍 방법을 사용할 수는 있으나 변수의 메모리 기능을 구현하기 위해 제작자는 새로운 프로그래밍 언어를 잘 알아야 하며, 제작된 멀티미디어 콘텐츠의 데이터 크기가 커지므로 네트워크를 통한 전송시 트래픽이 증가되며, 재생

시 그 프로그래밍 언어를 해석할 수 있는 해석 엔진이 있어야 한다. 재생 기기에 해석 엔진을 탑재해야 하는 점, 트래픽이 증가되는 점은 모바일 기기를 통하여 주고받을 수 있는 멀티미디어 콘텐츠의 요구가 커지고 있는 상황을 고려할 때 더욱 문제가 된다.

【발명이 이루고자 하는 기술적 과제】

<22> 따라서, 본 발명의 목적은 프로그래밍 방법을 사용하지 않고서도 SMIL을 사용하여 간단하고 쉽게 메모리 기능을 구현할 수 있도록 제작된 멀티미디어 콘텐츠, 그 제작 방법 및 재생 방법을 제공하는 것이다.

【발명의 구성 및 작용】

<23> 상기 목적은 본 발명에 따라, SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠에 있어서, 변수를 초기화하는 속성을 가진 제1 엘리먼트; 및 상기 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성을 가진 제2 엘리먼트를 포함하는 것을 특징으로 하는 멀티미디어 콘텐츠에 의해 달성된다.

<24> 상기 멀티미디어 콘텐츠는 상기 변수의 연산에 따른 결과를 표시하기 위한 제3 엘리먼트를 더 포함하고, 자신이 실행되기 위한 실행 조건을 표현하는 속성을 가질 수 있다.

<25> 상기 제2 엘리먼트의 속성은 상기 변수의 산술 연산, 관계 연산, 논리 연산 및 if 구문 중 적어도 하나를 표현하는 것이 바람직하고, 상기 변수의 연산이 처리되는 조건을 가지고 있음이 바람직하다.

<26> 상기 제2 엘리먼트의 변수의 연산은 상기 제2 엘리먼트가 파싱될 때 처리됨이 바람직하다.

- <27> 상기 제1 엘리먼트의 속성은 속성 이름 = '변수 = 0;'의 형태로 표현되고, 상기 제2 엘리먼트의 속성은 속성 이름 = '변수의 연산;'의 형태로 표현됨이 바람직하다.
- <28> 상기 제1 엘리먼트의 속성은 var = 'x = 0;' 형태로 표현되고, 상기 제2 엘리먼트의 속성은 var = 'x = x ◆ α' 형태로 표현되거나(◆는 산술연산자), 상기 제1 엘리먼트의 속성은 var = 'x = 0;' 형태로 표현되고, 상기 제2 엘리먼트의 속성은 var = 'count++; sum + = 20;' 형태로 표현될 수 있다.
- <29> 상기 제3 엘리먼트의 속성은 속성 이름 = '대응 태그의 실행 조건' 형태로 표현될 수 있으며, 특히 condition = '실행 조건' 형태로 표현될 수 있다.
- <30> 또한, 상기 목적은 SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠에 있어서, 적어도 하나의 엘리먼트; 및 상기 엘리먼트 중 적어도 하나에 관한 함수, 및 상기 함수가 참이 되었을 때 처리되어야 할 내용이 표현된 속성을 갖는 다른 엘리먼트를 포함하는 것을 특징으로 하는 멀티미디어 콘텐츠에 의해서도 달성된다.
- <31> 상기 함수는 대응하는 엘리먼트의 렌더링 영역이 소정 엘리먼트의 렌더링 영역과 서로 겹칠 때 참이 되고, 상기 다른 엘리먼트의 속성은 상기 함수가 참이 되면 대응하는 엘리먼트의 렌더링이 종료됨을 표현할 수 있다.
- <32> 상기 함수는 하나 이상의 엘리먼트들의 렌더링 영역이 서로 겹칠 때 참이 되고, 상기 다른 엘리먼트의 속성은 상기 함수가 참이 되면 대응하는 엘리먼트의 렌더링이 시작됨을 표현할 수 있다.
- <33> 한편, 본 발명의 다른 분야에 따르면, 상기 목적은 SMIL(Synchronized Multimedia Integration Language)을 사용하여 멀티미디어 콘텐츠를 제작하는 방법에 있어서, (a) 제1 엘

리먼트의 속성을 사용하여 변수를 초기화하는 단계; 및 (b) 제2 엘리먼트의 속성을 사용하여 상기 변수의 연산을 표현함으로써 메모리 기능을 구현하는 단계를 포함하는 것을 특징으로 하는 방법에 의해서도 달성된다.

- <34> 상기 방법은 (c) 제3 엘리먼트의 속성을 사용하여 상기 변수의 연산 결과에 따른 상기 제3 엘리먼트의 실행 조건을 표현하는 단계를 포함하는 것이 바람직하다.
- <35> 상기 (b)단계는 상기 변수의 산술 연산, 관계 연산, 논리 연산 및 if 구문 중 적어도 하나를 표현함으로써 메모리 기능을 구현하는 단계를 포함하는 것이 바람직하다. 또한, 상기 변수의 연산이 실행되기 위한 조건을 추가하는 단계를 더 포함하거나, SMIL에서 정의된 타이밍 관련 속성을 사용하여 상기 조건을 표현하는 단계를 포함하는 것이 바람직하고, 상기 (b)단계의 변수의 연산은 대응하는 엘리먼트가 파싱될 때 처리되는 것이 바람직하다.
- <36> 상기 (a)단계는 속성 이름 = '변수 = 0;'의 형태로 초기화하고, 상기 (b)단계는 속성 이름 = '변수의 연산;'의 형태로 표현하거나, 상기 (a)단계는 var = 'x = 0;' 형태로 초기화하고, 상기 (b)단계는 var = 'x = x ◆ a' 형태로 표현하는 것이 바람직하다(◆는 산술연산자).
- <37> 상기 (c)단계는 속성 이름 = '대응 태그의 실행 조건' 형태로 표현할 수 있으며, var = 'if 구문' 형태로 표현하는 것이 바람직하다.
- <38> 또한, 상기 목적은 SMIL(Synchronized Multimedia Integration Language)을 사용하여 멀티미디어 콘텐츠를 제작하는 방법에 있어서, (a) 엘리먼트 내에 적어도 하나의 함수를 정의하는 단계; 및 (b) 상기 함수가 참이 되었을 때 처리되어야 할 내용을 상기 엘리먼트의 속성으로 표현하는 단계를 포함하는 것을 특징으로 하는 방법에 의해서도 달성된다.
- <39> 상기 (b)단계는 상기 엘리먼트의 Action 속성으로 표현하는 것이 바람직하다.

- <40> 상기 (a)단계는 대응하는 엘리먼트의 렌더링 영역이 소정 엘리먼트의 렌더링 영역과 서로 겹칠 때 참이 되는 함수를 정의하는 단계를 포함하고, 상기 (b)단계는 상기 Action 속성을 사용하여 상기 대응하는 엘리먼트의 렌더링을 종료하거나, 상기 (a)단계는 하나 이상의 엘리먼트들의 렌더링 영역이 서로 겹칠 때 참이 되는 함수를 정의하는 단계를 포함하고, 상기 (b)단계는 상기 Action 속성을 사용하여 대응하는 엘리먼트의 렌더링을 시작함이 바람직하다.
- <41> 이하 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명한다.
- <42> 도 1은 본 발명의 바람직한 실시예에 따른 멀티미디어 콘텐츠를 재생하는 장치의 블럭도이다.
- <43> 도 1을 참조하면, 재생 장치는 본 발명에 따라 제작된 멀티미디어 콘텐츠인 SMIL 콘텐츠를 재생하기 위한 장치로서, SMIL 해석 엔진(1), 네트워크 통신부(2) 및 저장부(3)를 구비한다. SMIL 콘텐츠는 종래의 SMIL에 본 발명에 따라 변수의 메모리 기능을 정의하고 지원하는 속성이 추가된 SMIL로 작성된 멀티미디어 콘텐츠를 말한다. 다시 말해 SMIL 콘텐츠는 변수의 메모리 기능이 구현된 콘텐츠를 말한다. SMIL 콘텐츠에 대한 상세한 설명은 후술한다.
- <44> 네트워크 통신부(2)는 네트워크에 접속하여 SMIL 콘텐츠를 가져올 수 있다. 저장부(3)는 SMIL 콘텐츠를 저장한다. 저장부(3)는 하드디스크, 메모리, 광 디스크와 같은 정보저장매체로 구현된다. 저장부(3)가 광 디스크로 구현될 경우 광 디스크에 저장된 SMIL 콘텐츠를 읽어들이기 위해 저장부(3)는 광 디스크 드라이브(도시되지 않음)를 구비한다. 광 디스크는 광 디스크 드라이브에 착탈가능하도록 장착된다.

<45> SMIL 해석 엔진(1)은 SMIL 콘텐츠를 파싱하여 해석하고 실행하기 위한 엔진이다. 나아가, SMIL 해석 엔진(1)은 SMIL 콘텐츠에 포함된 오디오 파일, 이미지 파일, 동영상 파일 등 다양한 멀티미디어 파일을 디코딩하기 위한 디코더를 구비한다. 즉, SMIL 해석 엔진(1)은 네트워크 통신부(2) 또는 저장부(3)로부터의 SMIL 콘텐츠를 해석하고 실행하여 디스플레이 장치(도시되지 않음)에 디스플레이하는 한편 사용자 입력을 수신하여 그에 따른 동작을 수행한다. 특히, SMIL 해석 엔진(1)은 SMIL 콘텐츠 중 메모리 기능을 구현하는 변수의 연산을 정의하는 속성을 갖는 엘리먼트를 파싱하고, 이 때 엘리먼트의 실행에 대한 조건이 존재하면 그 조건이 만족되었을 때 이를 처리한다. 또한, 적어도 하나의 함수가 정의된 엘리먼트를 파싱하여 소정 속성에 표현된, 상기 함수가 참이 되었을 때 처리되어야 할 내용에 따라 처리한다. 이에, SMIL 콘텐츠에서의 사용자와의 인터랙션이 수행된다.

<46> 도 2는 본 발명의 바람직한 실시예에 따른 SMIL 콘텐츠의 개략도이다.

<47> 도 2를 참조하면, 본 실시예에 따른 SMIL 콘텐츠는 교육용 멀티미디어 콘텐츠로서, 초등학교 수학 학습을 위한 것으로, 그 전체 프레임의 상단에는 콘텐츠의 타이틀이 들어가는 영역이 마련되어 있고, 중단에는 학습 내용을 위한 영역이 마련되고 하단에는 학습 내용이 보여지는 동안 필요한 설명 및 안내 정보가 표시된다.

<48> 특히, 중단 프레임은 수학 학습을 위한 주요 진행 부분이 표시되는 영역으로서 내용과 문제를 표시한다. 다음 내용을 보고자 할 때 사용자는 '다음'버튼을 클릭한다.

<49> 도 3은 도 2의 프레임 구성을 위한 SMIL 콘텐츠의 개략을 보여준다.

- <50> 도 3을 참조하면, 전체 프레임은 <rootLayout...>으로 지정해주고, 타이틀, 학습 내용, 설명을 위한 영역은 각각 <region id='title' ...>, <region id='content' ...>, <region id='caption' ...>으로 정의하여 표현할 수 있다.
- <51> 도 4는 도 2의 SMIL 콘텐츠의 일 예를 보여준다.
- <52> 도 4를 참조하면, 중단 프레임의 첫번째 문제는 식별자 img1-1을 갖는 이미지가 정답인 경우를 나타내고 있다. 사용자가 그 이미지를 클릭하면 대응하는 이벤트가 발생되어 정답, 즉 식별자 rightAnswer를 가진 텍스트가 표시되고 나머지 정답이 아닌 이미지, 즉 식별자 img1-2, img1-3, img1-4를 갖는 이미지를 클릭하면 대응하는 이벤트가 발생되어 오답, 즉 식별자 wrong을 가진 텍스트가 표시된다. 두번째 문제는 식별자 img2-3을 갖는 이미지가 정답인 경우를 나타내고 있다. 사용자가 그 이미지를 클릭하면 대응하는 이벤트가 발생되어 정답, 즉 식별자 rightAnswer를 가진 텍스트가 표시되고 나머지 정답이 아닌 이미지, 즉 식별자 img2-1, img2-2, img2-4를 갖는 이미지를 클릭하면 대응하는 이벤트가 발생되어 오답, 즉 식별자 wrong을 가진 텍스트가 표시된다.
- <53> 특히, 도 4의 구현예에 따르면, 사용자가 맞춘 정답의 갯수를 누적하여 보여줄 수 있다. 즉, 첫번째 문제 부분의 엘리먼트 <image id='img1-1'../>에는 본 발명에 따라 정의된 변수를 초기화하는 속성으로 var = 'x=0;'이 삽입되어 있다. 또한, 첫번째 문제 부분의 엘리먼트 <text id='rightAnswer'../>에는 본 발명에 따라 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 var = 'x=x+1;'이 삽입되어 있고, 두번째 문제 부분의 엘리먼트 <text id='rightAnswer'../>에도 마찬가지로 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 var = 'x=x+1;'이 삽입되어 있다. 이에 따라, 사용자가 정답을 클릭하게 되면 그 횟수가 변수로서 누적된다. 따라서, 사용자에게 지금까지 맞춘 정답의 갯수를 알려줄 수 있게 된다.

<54> 도 5는 도 2의 SMIL 콘텐츠의 다른 예를 보여준다.

<55> 도 5의 구현예에 따르면, 각 문제마다 가중치를 부여할 수 있다. 즉, 첫번째 문제 부분의 엘리먼트 `<text id='rightAnswer'../>`에는 본 발명에 따라 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 `var = 'x=x+10;'`이 삽입되어 있고, 두번째 문제 부분의 엘리먼트 `<text id='rightAnswer'../>`에도 마찬가지로 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 `var = 'x=x+20;'`이 삽입되어 있다. 이에 따라, 사용자가 첫번째 문제의 정답을 클릭하게 되면 10점의 가중치가 부여되고, 두번째 문제에는 20점의 가중치가 부여된다.

<56> 도 6은 도 2의 SMIL 콘텐츠의 또 다른 예를 보여준다.

<57> 도 6의 구현예에 따르면, 틀린 문제에 대해 감점을 줄 수 있다. 즉, 첫번째 문제 부분의 엘리먼트 `<text id='wrong'../>`에는 본 발명에 따라 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 `var = 'x=x-5;'`가 삽입되어 있고, 두번째 문제 부분의 엘리먼트 `<text id='wrong'../>`에도 마찬가지로 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 `var = 'x=x-5;'`가 삽입되어 있다. 이에 따라, 사용자가 첫번째 문제와 두번째 문제의 오답을 클릭하게 되면 각각 5점이 감점된다.

<58> 도 7은 도 2의 SMIL 콘텐츠의 또 다른 예를 보여준다.

<59> 도 7의 구현예에 따르면, 도 4 내지 도 6의 예들을 조합함으로써, 정답의 갯수와 지금까지 획득한 점수를 각각 연산/저장하여 사용자에게 보여줄 수 있다. 즉, 첫번째 문제 부분의 엘리먼트 `<text id='rightAnswer'../>`에는 본 발명에 따라 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성으로 `var = 'count++; sum+=10;'`가 삽입되어 있고, 두번째 문제 부분의 엘리먼트 `<text id='rightAnswer'../>`에도 마찬가지로 변수의 연산을 표현함으로써 메모리 기능을

구현하는 속성으로 `var = 'count++; sum+=20;'`가 삽입되어 있다. 이에 따라, 사용자가 정답을 클릭한 횟수와 사용자가 획득한 점수가 동시에 누적되어 저장된다.

<60> 이처럼, 도 4 내지 도 7에 구현된 변수의 연산을 통한 메모리 기능에 의해 저장된 변수는 다양하게 활용될 수 있다.

<61> 도 8은 도 4 내지 도 7에 구현된 변수의 메모리 기능에 의해 저장된 변수를 활용하기 위한 예를 보여준다.

<62> 도 8을 참조하면, 식별자가 'good'인 텍스트 엘리먼트는 변수의 연산에 따른 결과를 표시하기 위한 엘리먼트로서, 사용자가 획득한 총 점수가 70이 초과된 경우에만 실행되어 대응하는 텍스트를 출력한다. 마찬가지로, 식별자가 'bad'인 텍스트 엘리먼트 또한 변수의 연산에 따른 결과를 표시하기 위한 엘리먼트로서, 사용자가 획득한 총 점수가 70 이하인 경우 실행되어 대응하는 텍스트를 출력한다.

<63> 한편, 도 4 내지 7에 구현된 기능을 수행하기 위한 변수의 연산은 일반적으로 대응하는 엘리먼트가 파싱될 때 처리되나, 그 실행 조건을 추가하여 변수의 연산에 대한 처리를 언제할 것인지를 조정할 수 있다.

<64> 도 9는 변수의 연산에 대한 처리 조건이 추가된 예를 보여준다.

<65> 도 9의 구현예에 따르면, `<image id='img1-1'.../>`은 자신이 실행되기 위한 실행 조건을 표현하는 속성 `var = 'if (activeEvent){x++;}'`을 가진 엘리먼트로서, 이에 의해, 사용자가 정답 또는 오답을 선택하였는지에 대한 결과를 출력하지는 않고 정답을 맞춘 개수만 기억하였다가 사용자가 획득한 총 점수만이 출력된다. 즉, 식별자 'img1-1'을 갖는 이미지가 클릭되었을 경우에만 변수의 값이 1 증가된다.

<66> 본 발명에 따라 정의된 SMIL의 새로운 속성과 그 내용을 정리하면 다음과 같다. [표 1]
은 속성을 나타내고 [표 2]는 속성의 상세 내용을 보여준다.

<67> 【표 1】

속성 이름	설명
var	변수의 연산 구분
condition	SMIL 엘리먼트의 실행 조건

<68> 【표 2】

구분	설명
산술연산	+, -, *, /
	%
	++, --
관계연산	>, <
	>=, <=
	=, !=
논리연산	!, &&,
if	변수 처리 조건

<69> 【표 3】

조건	설명
SMIL의 Timing 속성	begin, end, repeat SyncBase Timing
	beginEvent, endEvent, repeatEvent, activateEvent EventBase Timing

<70> if 처리 조건으로는 [표 3]와 같이 종래 SMIL에 정의된 엘리먼트의 Timing 속성과 [표 2]에 정의된 연산 속성을 가질 수 있다.

<71> 한편, 변수의 연산 기능이외에 함수 기능을 추가할 수 있다. 가령, 두 개의 애니메이션 이미지가 움직이는 중 충돌이 생기면 그 이미지가 없어지는 것을 표현하고자 할 때 종래 SMIL

에 따르면 이를 표현할 수 없다. 종래 SMIL은 [표 3]에 정의된 Timing 속성을 사용하여 일 이미지가 종료될 때 다른 이미지가 시작되는 것과 같이 그 대상 이미지가 하나인 경우를 처리할 수는 있지만 2 개의 이미지가 충돌하는 것과 같이 그 대상이 2 개 이상이면 처리할 수 없기 때문이다. 그러나, 본 발명에 따라 함수 기능을 추가하면 이와 같은 표현이 가능하다.

<72> 도 10은 본 발명에 따라 함수 기능이 추가된 일 예를 보여준다.

<73> 도 10을 참조하면, 렌더링 영역이 서로 겹칠 때 참이 되는 collision 함수가 정의되었다고 가정하였을 때, 식별자 'image1'인 이미지의 렌더링 영역과 엘리먼트 의 렌더링 영역의 collision이 일어나면 action 속성에 표현된 바에 자신, 즉 의 렌더링을 종료한다.

<74> 도 11은 본 발명에 따라 함수 기능이 추가된 다른 예를 보여준다.

<75> 도 11을 참조하면, 식별자 'image1'을 갖는 이미지와 식별자 'image2'를 갖는 이미지의 렌더링 영역이 서로 겹칠 때 식별자 'image3'을 갖는 이미지의 렌더링을 시작한다.

<76> 도 10 및 11의 예에서는 action 속성을 사용하여 처리하였던 것과 같이, 함수 기능은 그 함수의 조건을 만족할 경우, 즉 그 함수가 참이 될 경우에 대한 처리가 필요하다. 원하는 콘텐츠의 제작에 필요한 기본적인 함수의 내용이 정의되어 있다면 이들을 이용하여 적절하게 원하는 표현을 할 수 있다. 나아가, 콘텐츠를 종류 별로 묶거나 함수를 기능별로 묶어서 프로파 일이나 컴포넌트화하는 것도 가능하다. 또한, 정의된 함수들을 조합하여 새로운 함수를 정의하는 것도 가능하다.

<77> 상기와 같은 구성을 기초로 본 발명의 바람직한 실시예에 따른 멀티미디어 콘텐츠 제작 방법을 설명한다.

<78> 도 12는 본 발명의 일 실시예에 따라 속성이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트이다.

<79> 도 12를 참조하면, 본 발명에 따른 변수의 연산을 정의하여 메모리 기능을 구현하기 위해서는 먼저 제1 엘리먼트의 속성을 사용하여 변수를 초기화한다(1201단계). 다음으로, 제2 엘리먼트의 속성을 사용하여 초기화된 변수의 연산을 표현함으로써 메모리 기능을 구현한다(1202단계). 변수의 연산은 산술 연산, 관계 연산, 논리 연산 및 if 구문 중 적어도 하나로 표현할 수 있다. 또한, 제3 엘리먼트의 속성을 사용하여 상기 변수의 연산 결과에 따른 제3 엘리먼트의 실행 조건을 표현하여 변수 연산 결과를 소정 목적에 맞게 활용한다(1203단계). 그 예로는 도 8을 참조하여 전술한 바 있다.

<80> 도 13은 본 발명의 다른 실시예에 따라 속성이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트이다.

<81> 도 13을 참조하면, 본 발명에 따른 변수의 연산을 정의하여 메모리 기능을 구현하기 위해, 먼저 속성 이름 = '변수 = 0;'의 형태로 변수를 초기화한다(1301단계). 가령 var = 'x = 0;' 형태로 초기화된다. 다음으로, 속성 이름 = '변수의 연산;'의 형태로 변수의 연산을 표현한다(1302단계). 예로, var = 'x = x ◆ a' 형태로 표현할 수 있다(◆는 산술연산자). 또한, 변수의 연산이 실행되기 위한 조건을 속성 이름 = '대응 태그의 실행 조건' 형태로 표현한다(1303단계). 가령, var = 'if 구문' 형태로 표현할 수 있다.

<82> 도 14는 본 발명의 일 실시예에 따라 함수 기능이 추가된 SMIL을 사용하여 멀티미디어 콘텐츠를 제작하는 방법을 설명하기 위한 플로우차트이다.

<83> 도 14를 참조하면, 적어도 하나의 함수를 정의한다(1401단계). 일 예로, 대응하는 엘리먼트의 렌더링 영역이 소정 엘리먼트의 렌더링 영역과 서로 겹칠 때 참이 되는 함수를 정의하거나 하나 이상의 엘리먼트들의 렌더링 영역이 서로 겹칠 때 참이 되는 함수를 정의한다. 다음으로, 정의된 함수가 참이 되었을 때 처리되어야 할 내용을 소정 엘리먼트의 속성으로 표현한다(1402단계). 일 예로 Action 속성을 사용하여 처리되어야 할 내용을 표현할 수 있다. 즉, Action 속성을 이용하여 대응하는 엘리먼트의 렌더링을 종료하거나 시작할 수 있다.

【발명의 효과】

<84> 전술한 바와 같이, 본 발명에 따른 SMIL 콘텐츠는 변수의 연산에 따른 메모리 기능을 구비함으로 인해 서버와의 접속없이도 콘텐츠만으로 사용자 입력에 대한 결과를 누적하거나 저장하여 보여줄 수 있다. 다시 말해, 프로그래밍 작업을 하지 않고 SMIL에 새롭게 정의된 속성을 사용하여 간단하게 메모리 기능을 구현할 수 있다. 이와 같은 기능은 특히 모바일 환경에서 보다 효과적이다. 왜냐하면 모바일 환경에서는 자원의 제약이 적지 않기 때문이다. 별개의 프로그래밍 언어를 모르는 개발자라도 SMIL을 통해 사용자 입력에 보다 다양하고 직접적으로 반응할 수 있는 콘텐츠를 제작할 수 있다.

【특허청구범위】**【청구항 1】**

SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠에 있어서,

변수를 초기화하는 속성을 가진 제1 엘리먼트; 및

상기 변수의 연산을 표현함으로써 메모리 기능을 구현하는 속성을 가진 제2 엘리먼트를 포함하는 것을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 2】

제1항에 있어서,

상기 변수의 연산에 따른 결과를 표시하기 위한 제3 엘리먼트를 포함하는 것을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 3】

제2항에 있어서,

상기 제3 엘리먼트는 자신이 실행되기 위한 실행 조건을 표현하는 속성을 가짐을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 4】

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 제2 엘리먼트의 속성은 상기 변수의 산술 연산, 관계 연산, 논리 연산 및 if 구문 중 적어도 하나를 표현함을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 5】

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 제2 엘리먼트의 속성은 상기 변수의 연산이 처리되는 조건을 가지고 있음을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 6】

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 제2 엘리먼트의 변수의 연산은 상기 제2 엘리먼트가 파싱될 때 처리됨을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 7】

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 제1 엘리먼트의 속성은 속성 이름 = '변수 = 0;'의 형태로 표현되고,

상기 제2 엘리먼트의 속성은 속성 이름 = '변수의 연산;'의 형태로 표현됨을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 8】

제7항에 있어서,

상기 제1 엘리먼트의 속성은 var = 'x = 0;' 형태로 표현되고,

상기 제2 엘리먼트의 속성은 var = 'x = x ◆ a' 형태로 표현됨을 특징으로 하는 멀티미디어 콘텐츠(◆는 산술연산자).

【청구항 9】

제7항에 있어서,

상기 제1 엘리먼트의 속성은 `var = 'x = 0;'` 형태로 표현되고,

상기 제2 엘리먼트의 속성은 `var = 'count++; sum + = 20;'` 형태로 표현됨을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 10】

제8항 또는 제9항에 있어서,

상기 제3 엘리먼트의 속성은 속성 이름 = '대응 태그의 실행 조건' 형태로 표현됨을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 11】

제10항에 있어서,

상기 제3 엘리먼트의 속성은 `condition = '실행 조건'` 형태로 표현됨을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 12】

SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠에 있어서,

적어도 하나의 엘리먼트; 및

상기 엘리먼트 중 적어도 하나에 관한 함수, 및 상기 함수가 참이 되었을 때 처리되어야 할 내용이 표현된 속성을 갖는 다른 엘리먼트를 포함하는 것을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 13】

제12항에 있어서,

상기 함수는 대응하는 엘리먼트의 렌더링 영역이 소정 엘리먼트의 렌더링 영역과 서로 겹칠 때 참이 되고,

상기 다른 엘리먼트의 속성은 상기 함수가 참이 되면 대응하는 엘리먼트의 렌더링이 종료됨을 표현하는 것을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 14】

제13항에 있어서,

상기 함수는 하나 이상의 엘리먼트들의 렌더링 영역이 서로 겹칠 때 참이 되고,

상기 다른 엘리먼트의 속성은 상기 함수가 참이 되면 대응하는 엘리먼트의 렌더링이 시작됨을 표현하는 것을 특징으로 하는 멀티미디어 콘텐츠.

【청구항 15】

SMIL(Synchronized Multimedia Integration Language)을 사용하여 멀티미디어 콘텐츠를 제작하는 방법에 있어서,

(a) 제1 엘리먼트의 속성을 사용하여 변수를 초기화하는 단계; 및

(b) 제2 엘리먼트의 속성을 사용하여 상기 변수의 연산을 표현함으로써 메모리 기능을 구현하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 16】

제15항에 있어서,

(c) 제3 엘리먼트의 속성을 사용하여 상기 변수의 연산 결과에 따른 상기 제3 엘리먼트의 실행 조건을 표현하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 17】

제15항 또는 제16항에 있어서,

상기 (b)단계는 상기 변수의 산술 연산, 관계 연산, 논리 연산 및 if 구문 중 적어도 하나를 표현함으로써 메모리 기능을 구현하는 단계임을 특징으로 하는 방법.

【청구항 18】

제15항 또는 제16항에 있어서,

상기 (b)단계는 상기 변수의 연산이 실행되기 위한 조건을 추가하는 단계를 더 포함하는 것을 특징으로 하는 방법.

【청구항 19】

제18항에 있어서,

상기 (b)단계는

SMIL에서 정의된 타이밍 관련 속성을 사용하여 상기 조건을 표현하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 20】

제15항 또는 제16항에 있어서,

상기 (a)단계는 `var = 'x = 0;'` 형태로 초기화하고,

상기 (b)단계는 `var = 'x = x ◆ a'` 형태로 표현함을 특징으로 하는 방법(◆는 산술연산자).

【청구항 21】

SMIL(Synchronized Multimedia Integration Language)을 사용하여 멀티미디어 콘텐츠를 제작하는 방법에 있어서,

- (a) 엘리먼트 내에 적어도 하나의 함수를 정의하는 단계; 및
- (b) 상기 함수가 참이 되었을 때 처리되어야 할 내용을 상기 엘리먼트의 속성으로 표현하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 22】

제21항에 있어서,

상기 (b)단계는 상기 엘리먼트의 Action 속성으로 표현하는 단계임을 특징으로 하는 방법.

【청구항 23】

제21항에 있어서,

상기 (a)단계는 대응하는 엘리먼트의 렌더링 영역이 소정 엘리먼트의 렌더링 영역과 서로 겹칠 때 참이 되는 함수를 정의하는 단계를 포함하고,

상기 (b)단계는 상기 Action 속성을 사용하여 상기 대응하는 엘리먼트의 렌더링을 종료함을 특징으로 하는 방법.

【청구항 24】

제21항에 있어서,

상기 (a)단계는 하나 이상의 엘리먼트들의 렌더링 영역이 서로 겹칠 때 참이 되는 함수를 정의하는 단계를 포함하고,

상기 (b)단계는 상기 Action 속성을 사용하여 대응하는 엘리먼트의 렌더링을 시작함을 특징으로 하는 방법.

【청구항 25】

SMIL(Synchronized Multimedia Integration Language)을 사용하여 제작된 멀티미디어 콘텐츠스를 재생하는 방법에 있어서,

(a) 메모리 기능을 구현하는 변수의 연산을 정의하는 속성을 갖는 엘리먼트를 파싱하여 처리하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 26】

제25항에 있어서,

상기 (a)단계는

상기 엘리먼트의 실행에 대한 조건이 존재하면 그 조건이 만족되었을 때 상기 엘리먼트를 파싱하여 처리하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 27】

제25항 또는 제26항에 있어서,

(b) 적어도 하나의 함수가 정의된 엘리먼트를 파싱하여 소정 속성에 표현된, 상기 함수가 참이 되었을 때 처리되어야 할 내용에 따라 처리하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 28】

제27항에 있어서,

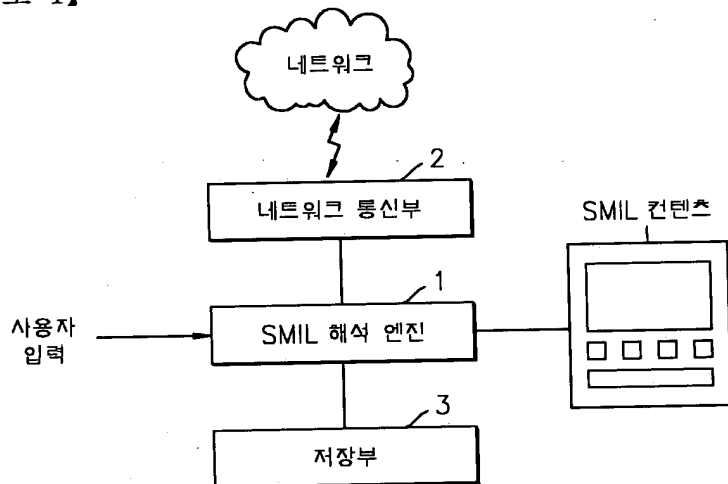
1020020074641

출력 일자: 2003/8/1

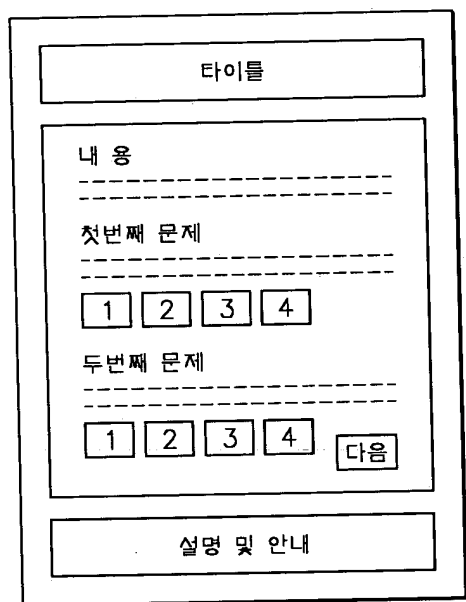
상기 (b)단계는 action 속성으로 표현된 내용에 따라 처리하는 단계를 포함하는 것을 특징으로 하는 방법.

【도면】

【도 1】



【도 2】



【도 3】

```
<rootLayout>
<region id="title"...>
  :
</region>
<region id="content"...>
  :
</region>
<region id="caption"...>
  :
</region>
</rootLayout>
```

【도 4】

```
<par> // 첫번째 문제 부분
  <image id="img1-1"
    var="x=0;".../>
  <image id="img1-2".../>
  <image id="img1-3".../>
  <image id="img1-4".../>
  <text id="rightAnswer"...
    var="x=x+1;"
    begin="img1-1.activateEvent"
    .../>
  <text id="wrong"...
    begin="img1-2.activateEvent;
    img1-3.activateEvent;
    img1-4.activateEvent".../>
</par>
<par> // 두번째 문제 부분
  <image id="img2-1".../>
  <image id="img2-2".../>
  <image id="img2-3".../>
  <image id="img2-4".../>
  <text id="rightAnswer"...
    var="x=x+1;"
    begin="img2-3.activateEvent"
    .../>
  <text id="wrong"...
    begin="img2-1.activateEvent;
    img2-2.activateEvent"
    img2-4.activateEvent".../>
</par>
```

⋮

【도 5】

```
<par> // 첫번째 문제 부분
...
<text id="rightAnswer"...
  var="x=x+10;"
  begin="img1-1.activateEvent"
.../>
...
</par>
<par> // 두번째 문제 부분
...
<text id="rightAnswer"...
  var="x=x+20;"
  begin="img2-3.activateEvent"
.../>
</par>
```

⋮

【도 6】

```
<par> // 첫번째 문제 부분
...
<text id="rightAnswer"...
  var="x=x+10;"
  begin="img1-1.activateEvent"
.../>
<text id="wrong"...
  var="x=x-5;"
  begin="img2.activateEvent;
  img3.activateEvent;
  img4.activateEvent".../>
...
</par>
<par> // 두번째 문제 부분
...
<text id="rightAnswer"...
  var="x=x+20;"
  begin="img2-3.activateEvent"
.../>
<text id="wrong"...
  var="x=x-5;"
  begin="img2-1.activateEvent;
  img2-2.activateEvent;
  img2-4.activateEvent".../>
</par>
```

⋮

【도 7】

```
<par> // 첫번째 문제 부분
...
<text id="rightAnswer"...
  var="count++; sum+=10;"
  begin="img1-1.activateEvent"
  .../>
...
</par>
<par> // 두번째 문제 부분
...
<text id="rightAnswer"...
  var="count++; sum+=20;"
  begin="img2-3.activateEvent"
  .../>
</par>
```

⋮

【도 8】

```
<par> // 최종 결과 부분
<text id="good"...
  condition="sum>70" .../>
<text id="bad"...
  condition="sum<=70" .../>
</par>
```

⋮

【도 9】

```
<par>
  <text id="img1-1"
    var="if (activeEvent) {x++;}
    .../>
  <image id="img1-2" .../>
  <image id="img1-3" .../>
  <image id="img1-4" .../>
</par>
```

⋮

【도 10】

```
<img ...
  function="collision(image1)
  action="end" .../>
```

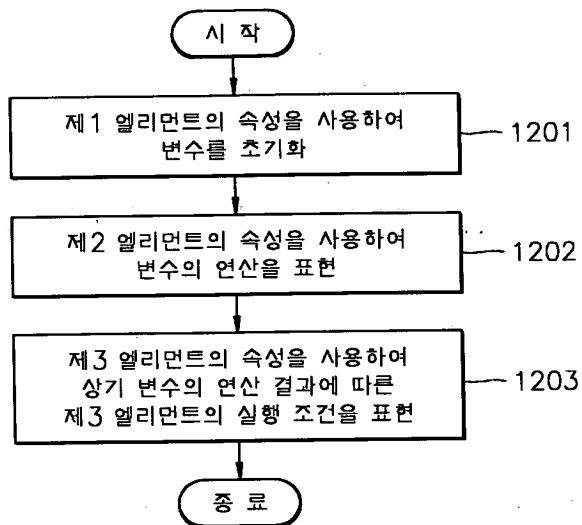
⋮

【도 11】

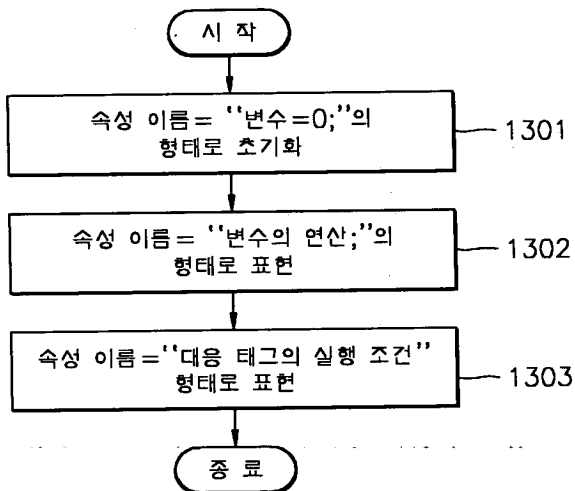
```
<img ...
  function="collision(image1,
    image2)
  action="image3.begin" .../>
```

⋮

【도 12】



【도 13】



【도 14】

